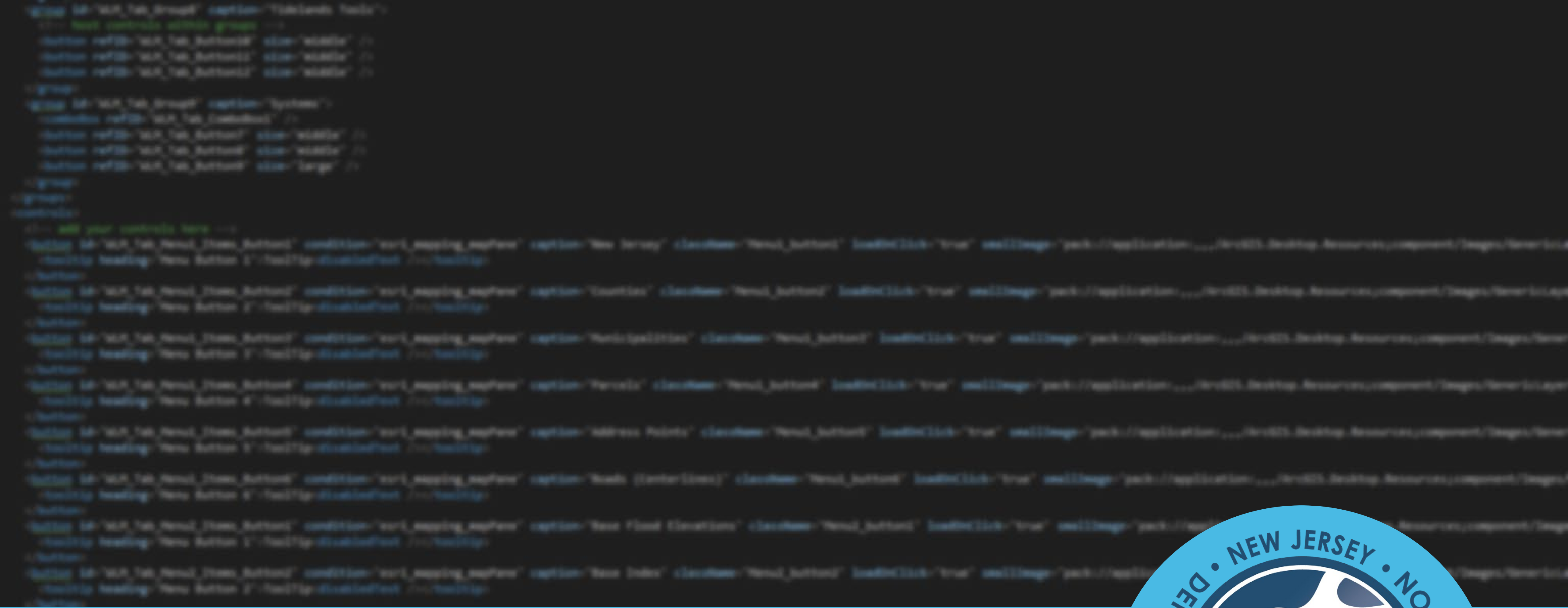


The Button That Doesn't Work

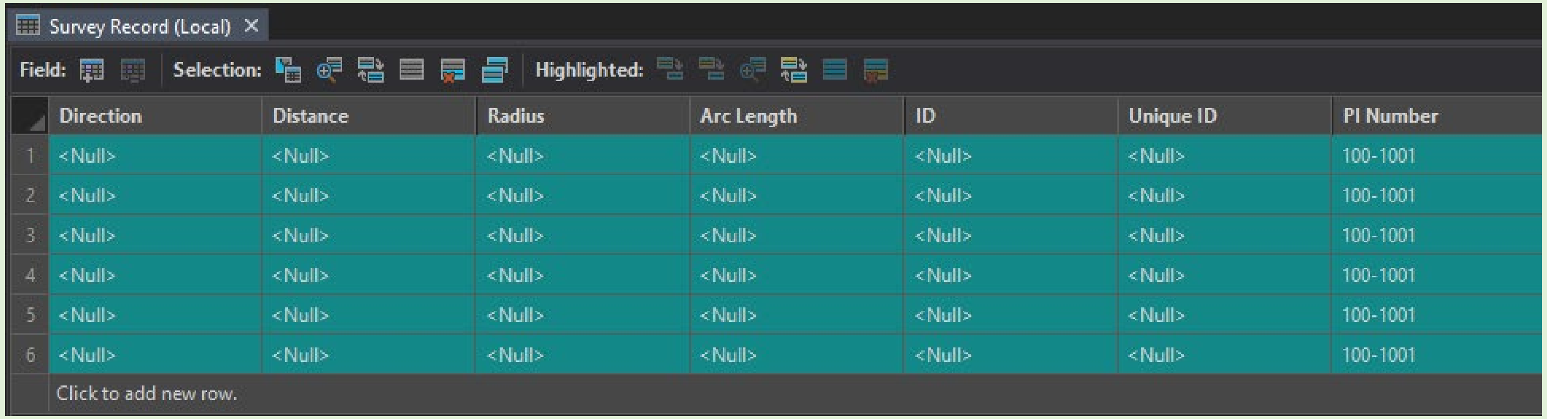
Andrew C. Garcia - NJDEP

Background

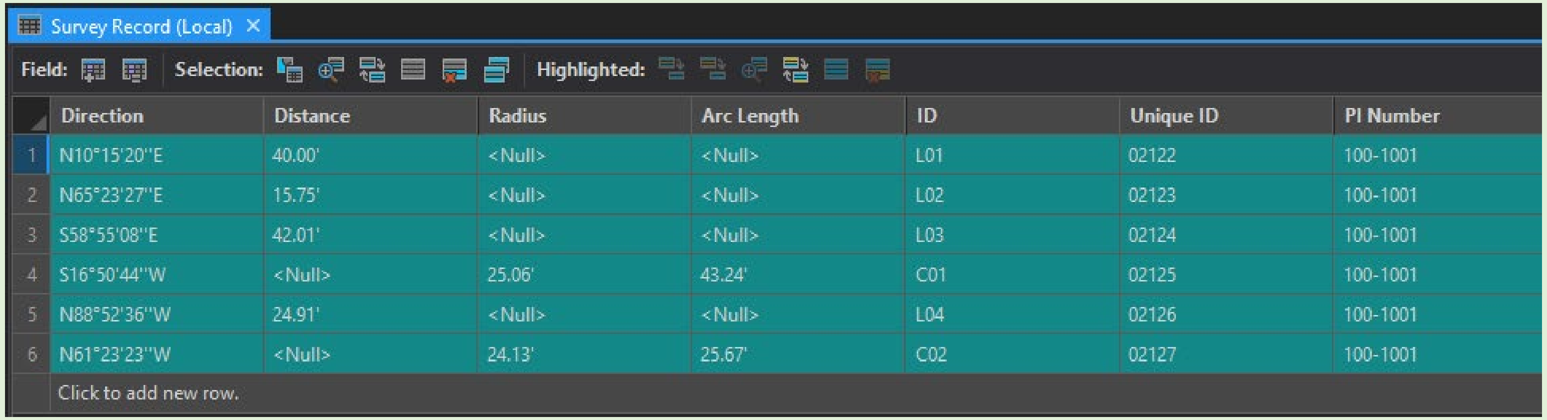
I need to populate a table in a drawing with dynamic text. The table contains values for metes and bounds of COGO linework.

Purpose

Streamline editing of values in a layer to reduce errors. Create a custom button using ArcGIS Pro Software Development Kit (SDK). Have that button perform field calculations on a specific layer.



	Direction	Distance	Radius	Arc Length	ID	Unique ID	PI Number
1	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
2	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
3	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
4	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
5	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
6	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	100-1001
	Click to add new row.						



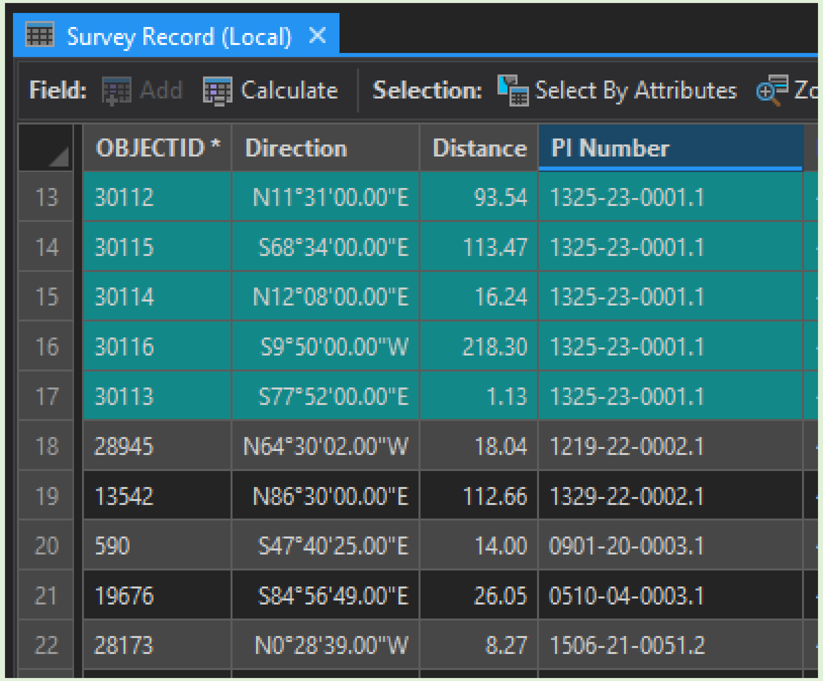
	Direction	Distance	Radius	Arc Length	ID	Unique ID	PI Number
1	N10°15'20"E	40.00'	<Null>	<Null>	L01	02122	100-1001
2	N65°23'27"E	15.75'	<Null>	<Null>	L02	02123	100-1001
3	S58°55'08"E	42.01'	<Null>	<Null>	L03	02124	100-1001
4	S16°50'44"W	<Null>	25.06'	43.24'	C01	02125	100-1001
5	N88°52'36"W	24.91'	<Null>	<Null>	L04	02126	100-1001
6	N61°23'23"W	<Null>	24.13'	25.67'	C02	02127	100-1001
	Click to add new row.						

Challenges

Make sure the button only runs when a specific layer is selected.

Make sure the button doesn’t run when nothing is selected (yes, there’s a difference).

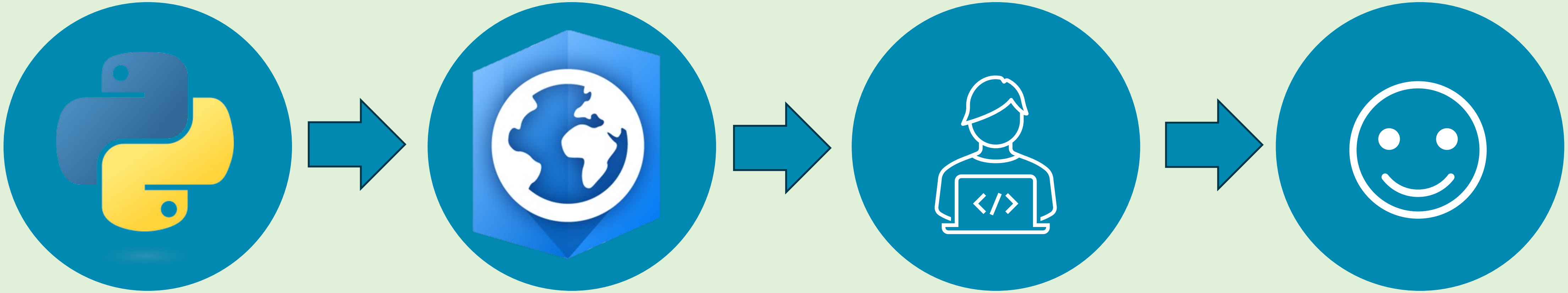
Make sure the button only runs when certain features within the layer are selected.



	OBJECTID *	Direction	Distance	PI Number
13	30112	N11°31'00.00"E	93.54	1325-23-0001.1
14	30115	S68°34'00.00"E	113.47	1325-23-0001.1
15	30114	N12°08'00.00"E	16.24	1325-23-0001.1
16	30116	S9°50'00.00"W	218.30	1325-23-0001.1
17	30113	S77°52'00.00"E	1.13	1325-23-0001.1
18	28945	N64°30'02.00"W	18.04	1219-22-0002.1
19	13542	N86°30'00.00"E	112.66	1329-22-0002.1
20	590	S47°40'25.00"E	14.00	0901-20-0003.1
21	19676	S84°56'49.00"E	26.05	0510-04-0003.1
22	28173	N0°28'39.00"W	8.27	1506-21-0051.2

Workflow

The making of the button that doesn't work.



Create an ArcPy script that runs the field calculations on the desired layer.

Create a new button in a custom tab using C# in ArcGIS Pro SDK.

Configure module script to only turn the button on when a feature of a specific layer is selected.

Furiously search StackExchange and see that someone resolved your issue years ago.

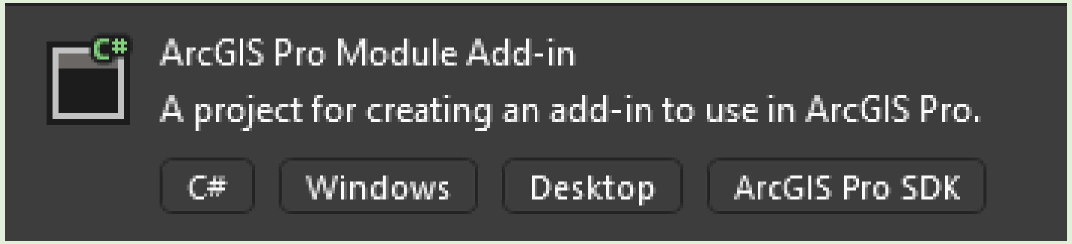
Setting up a custom tab

Install Visual Studio 2022

Install .NET Framework 4.8 Developer Pack

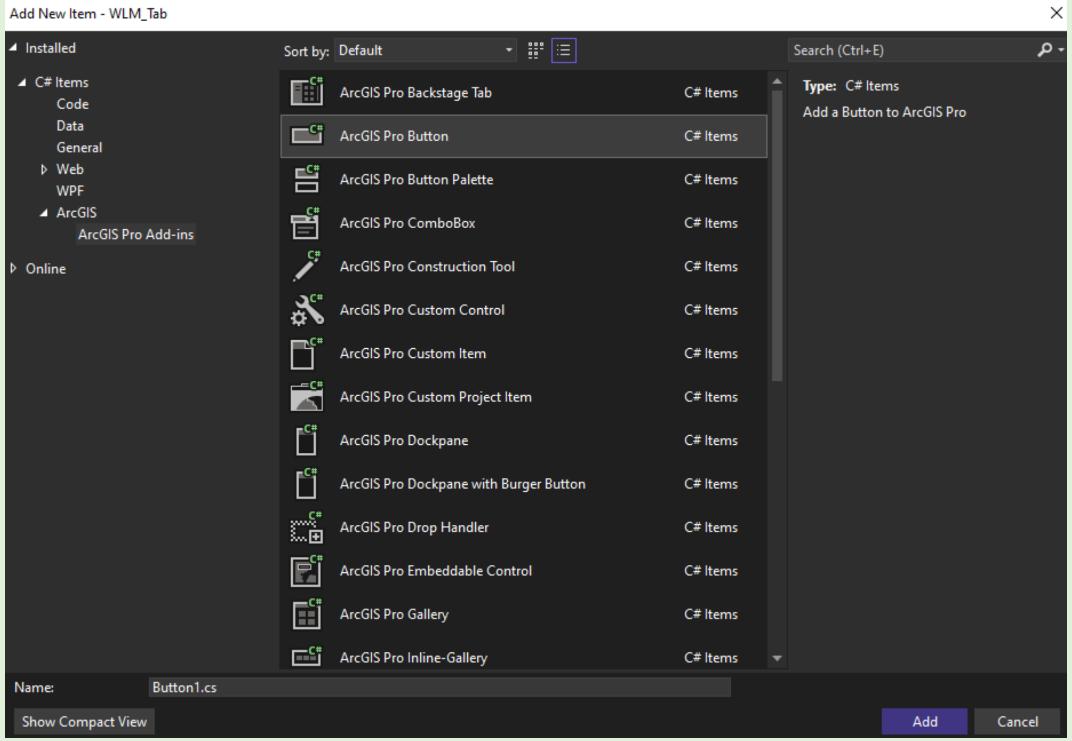
Install ArcGIS Pro SDK for .NET

Make sure the version of the SDK you install is the same as your ArcPro version. For this tab we are running ArcGIS Pro version 3.2.0.



ArcGIS Pro Module Add-in

A project for creating an add-in to use in ArcGIS Pro.



Add New Item - WLM_Tab

ArcGIS Pro Backstage Tab	C# Items
ArcGIS Pro Button	C# Items
ArcGIS Pro Button Palette	C# Items
ArcGIS Pro ComboBox	C# Items
ArcGIS Pro Construction Tool	C# Items
ArcGIS Pro Custom Control	C# Items
ArcGIS Pro Custom Item	C# Items
ArcGIS Pro Custom Project Item	C# Items
ArcGIS Pro Dockpane	C# Items
ArcGIS Pro Dockpane with Burger Button	C# Items
ArcGIS Pro Drop Handler	C# Items
ArcGIS Pro Embeddable Control	C# Items
ArcGIS Pro Gallery	C# Items
ArcGIS Pro Inline-Gallery	C# Items

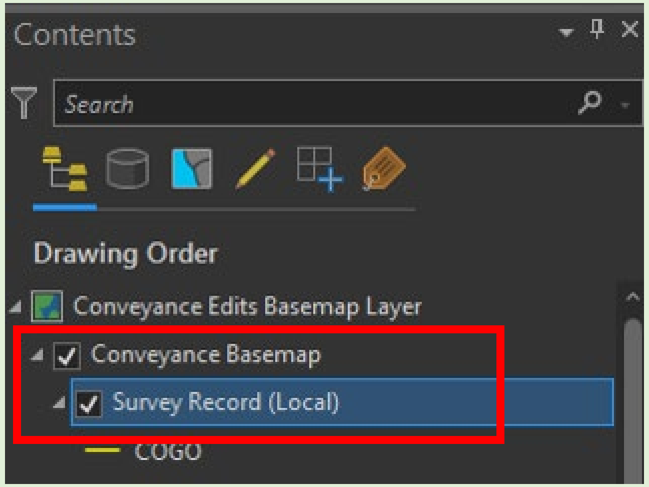
Type: C# Items

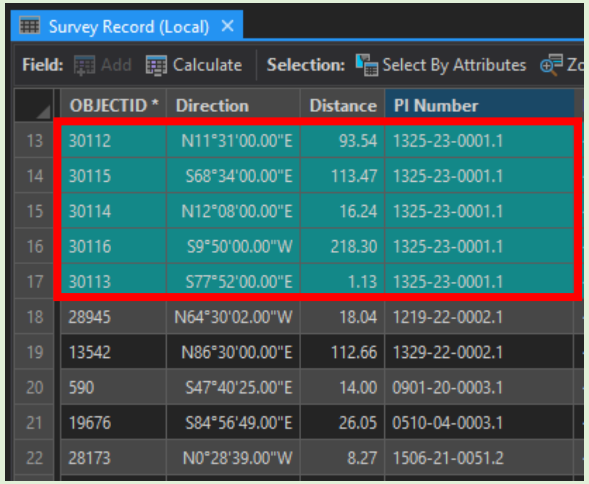
Add a Button to ArcGIS Pro

Name: Button1.cs

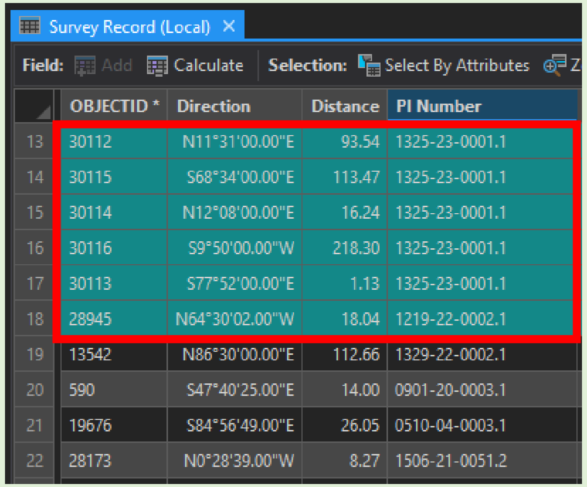
Python Functions

```
        except:
            arcpy.AddError("Could not calculate line distance.")

        #Calculate curve distance to add 2 sigfigs, foot tick mark, and thousands seperator.
        try:
            curvedistance = ["reference_document","radius","hyperlink","arclength"]
            radiuscommapos = '"radius" IS NOT NULL AND CHAR_LENGTH("reference_document")>=8 AND "radius">=0'
            radiuscommaneg = '"radius" IS NOT NULL AND CHAR_LENGTH("reference_document")>=9 AND "radius"<0'
            arclengthcomma = '"radius" IS NOT NULL AND CHAR_LENGTH("hyperlink")>=8'
            with arcpy.da.UpdateCursor(layer,curvedistance,curve) as cursor:
                for row in cursor:
                    row[0] = str('{:,.2f}'.format(row[1]))+"\'"
                    row[2] = str('{:,.2f}'.format(row[3]))+"\'"
                    cursor.updateRow(row)
            with arcpy.da.UpdateCursor(layer,curvedistance,radiuscommapos) as cursor:
                for row in cursor:
                    row[0]  = row[0][:-7]+','+row[0][-7:]
                    cursor.updateRow(row)
            with arcpy.da.UpdateCursor(layer,curvedistance,radiuscommaneg) as cursor:
                for row in cursor:
                    row[0]  = row[0][:-7]+','+row[0][-7:]
                    cursor.updateRow(row)
            with arcpy.da.UpdateCursor(layer,curvedistance,arclengthcomma) as cursor:
                for row in cursor:
                    row[2]  = row[2][:-7]+','+row[2][-7:]
                    cursor.updateRow(row)
        except:
            arcpy.AddError("Could not calculate curve distance.")

        #Calculate direction using cardinal directions. Note: the script below is written in arcade.
        try:
            arcpy.management.CalculateField(layer, "description", """// Change the settings portion to configure direction format, color, rounding and abbreviations
            // This is an Arcade expression

            // SETTINGS
            var QuadrantBearingFormat = true; //set 'true' for quadrant bearing, 'false' for north azimuth
            var ShowDistance = true;   //set as 'true' to show distance
            var ShowDirection = true;  //set as 'true' to show direction
            var ShowRadius
            var ShowCurvePa
            var CurveParam
            var ErrorString
            var RadiusAbbr
            var ArclengthAb
            var ChordAbbr =
            var AngleAbbr =
            var DistUnitAbb
            var NumberForma

            // VARIABLES
            var direction=$feature.Direction;
            var distance=$feature.Distance;
```





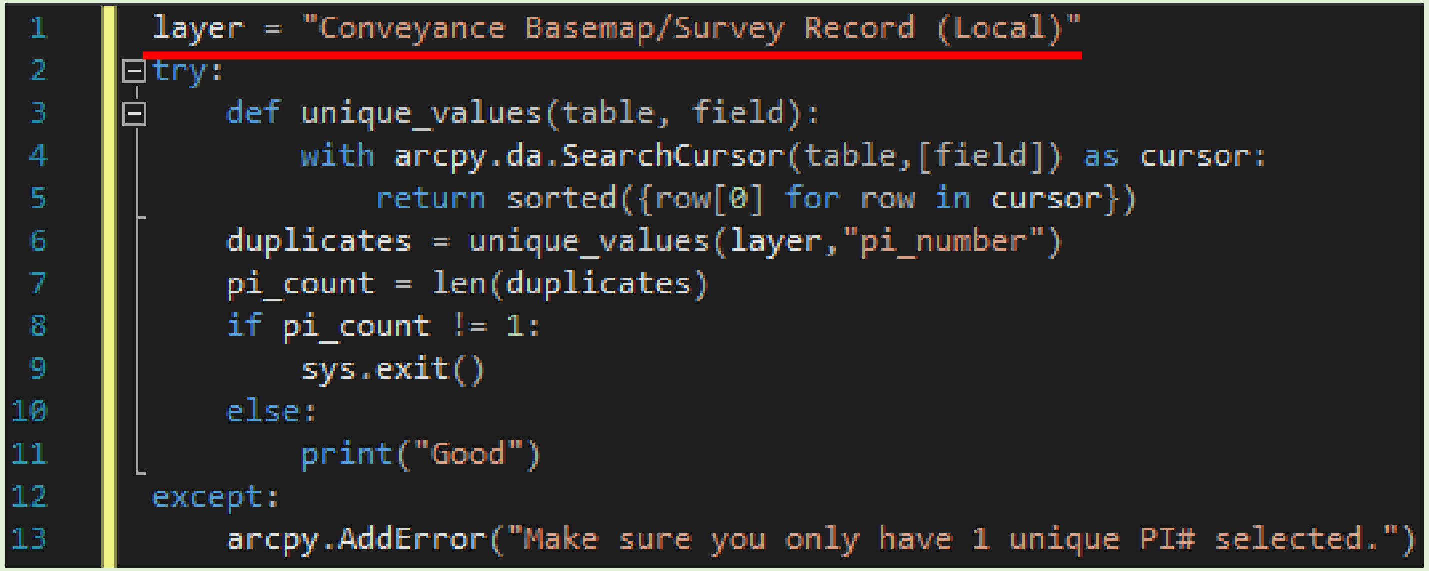
	OBJECTID *	Direction	Distance	PI Number
13	30112	N11°31'00.00"E	93.54	1325-23-0001.1
14	30115	S68°34'00.00"E	113.47	1325-23-0001.1
15	30114	N12°08'00.00"E	16.24	1325-23-0001.1
16	30116	S9°50'00.00"W	218.30	1325-23-0001.1
17	30113	S77°52'00.00"E	1.13	1325-23-0001.1
18	28945	N64°30'02.00"W	18.04	1219-22-0002.1
19	13542	N86°30'00.00"E	112.66	1329-22-0002.1
20	590	S47°40'25.00"E	14.00	0901-20-0003.1
21	19676	S84°56'49.00"E	26.05	0510-04-0003.1
22	28173	N0°28'39.00"W	8.27	1506-21-0051.2



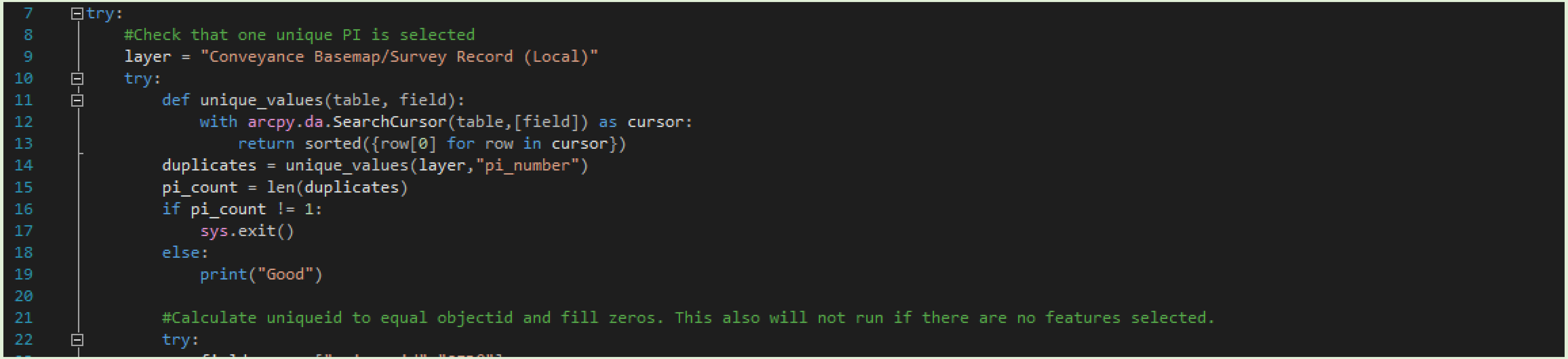
	OBJECTID *	Direction	Distance	PI Number
13	30112	N11°31'00.00"E	93.54	1325-23-0001.1
14	30115	S68°34'00.00"E	113.47	1325-23-0001.1
15	30114	N12°08'00.00"E	16.24	1325-23-0001.1
16	30116	S9°50'00.00"W	218.30	1325-23-0001.1
17	30113	S77°52'00.00"E	1.13	1325-23-0001.1
18	28945	N64°30'02.00"W	18.04	1219-22-0002.1
19	13542	N86°30'00.00"E	112.66	1329-22-0002.1
20	590	S47°40'25.00"E	14.00	0901-20-0003.1
21	19676	S84°56'49.00"E	26.05	0510-04-0003.1
22	28173	N0°28'39.00"W	8.27	1506-21-0051.2

Nesting Try/Except Functions

Try and Except functions are used for error handling. Errors will appear in ArcPro if the script were to fail.



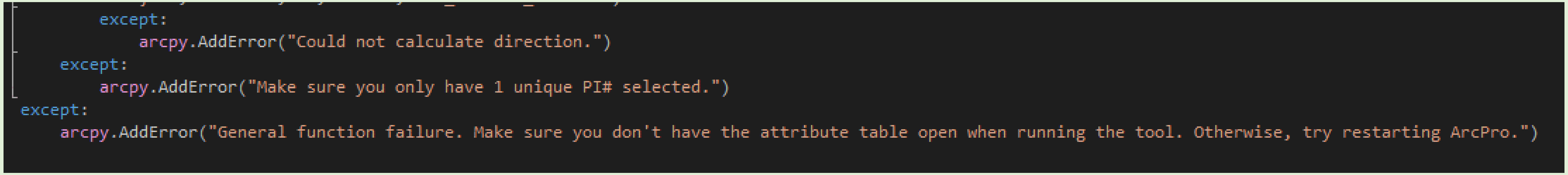
```python
layer = "Conveyance Basemap/Survey Record (Local)"
try:
    def unique_values(table, field):
        with arcpy.da.SearchCursor(table,[field]) as cursor:
            return sorted({row[0] for row in cursor})
    duplicates = unique_values(layer,"pi_number")
    pi_count = len(duplicates)
    if pi_count != 1:
        sys.exit()
    else:
        print("Good")
except:
    arcpy.AddError("Make sure you only have 1 unique PI# selected.")
```



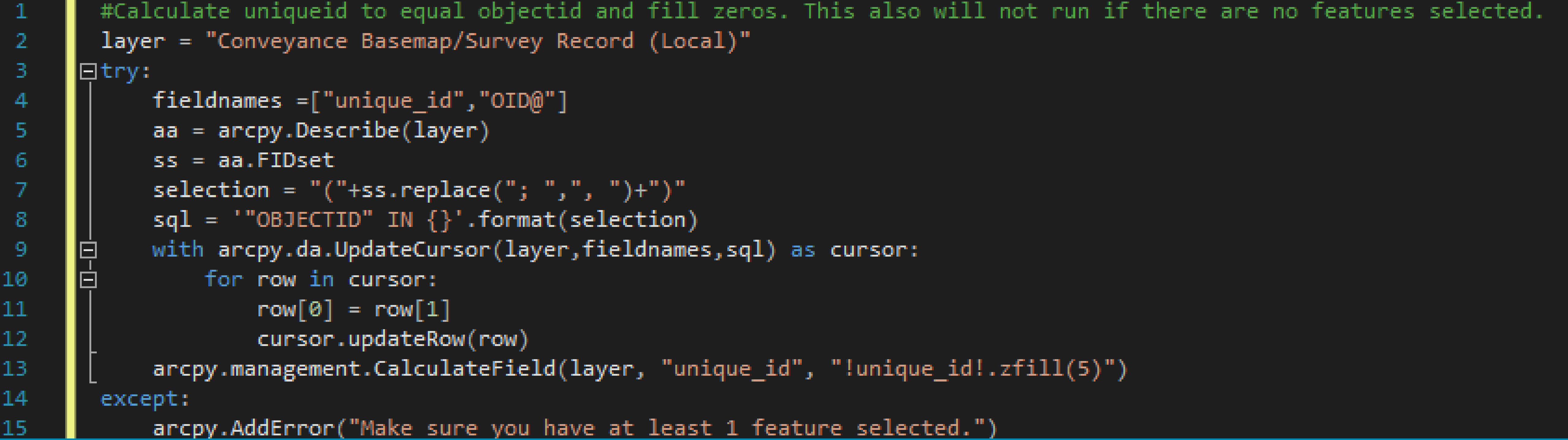
```
try:
    #Check that one unique PI is selected
    layer = "Conveyance Basemap/Survey Record (Local)"
    try:
        def unique_values(table, field):
            with arcpy.da.SearchCursor(table,[field]) as cursor:
                return sorted({row[0] for row in cursor})
        duplicates = unique_values(layer,"pi_number")
        pi_count = len(duplicates)
        if pi_count != 1:
            sys.exit()
        else:
            print("Good")

        #Calculate uniqueid to equal objectid and fill zeros. This also will not run if there are no features selected.
        try:
```

Nesting Try/Except Functions

Try and Except functions are used for error handling. Errors will appear in ArcPro if the script were to fail.



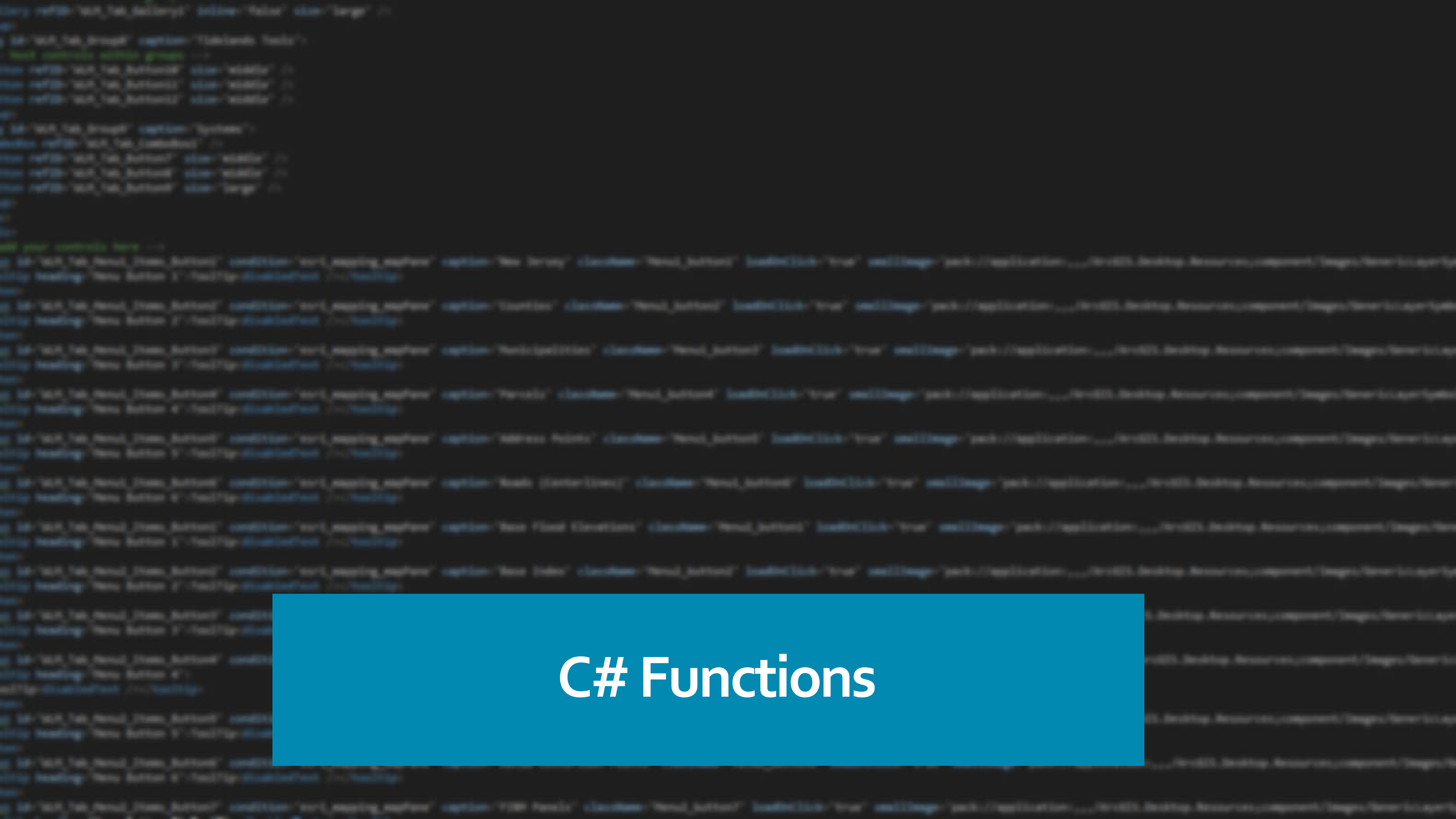
```
        except:
            arcpy.AddError("Could not calculate direction.")
    except:
        arcpy.AddError("Make sure you only have 1 unique PI# selected.")
except:
    arcpy.AddError("General function failure. Make sure you don't have the attribute table open when running the tool. Otherwise, try restarting ArcPro.")
```



```
#Calculate uniqueid to equal objectid and fill zeros. This also will not run if there are no features selected.
layer = "Conveyance Basemap/Survey Record (Local)"
try:
    fieldnames =["unique_id","OID@"]
    aa = arcpy.Describe(layer)
    ss = aa.FIDset
    selection = "("+ss.replace("; ",", ")+")"
    sql = '"OBJECTID" IN {}'.format(selection)
    with arcpy.da.UpdateCursor(layer,fieldnames,sql) as cursor:
        for row in cursor:
            row[0] = row[1]
            cursor.updateRow(row)
    arcpy.management.CalculateField(layer, "unique_id", "!unique_id!.zfill(5)")
except:
    arcpy.AddError("Make sure you have at least 1 feature selected.")
```

Redundant Functions

arcpy.management.CalculateField	arcpy.da.UpdateCursor
Calculates field based on set parameters.	Calculates field based on set parameters.
Runs only on selected features.	Runs only on selected features.
Runs even when there are no features selected.	Can set SQL statement to not run when there are no features selected.



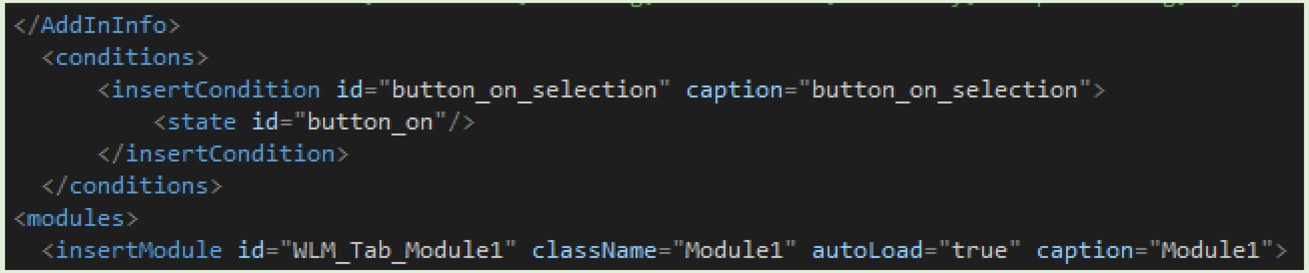
C# Functions

Creating a condition

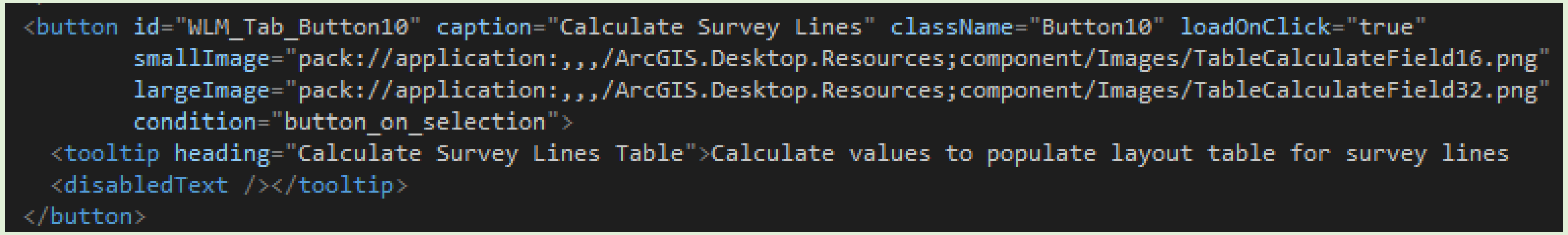
Condition is created inside the config.daml.

It is then called inside the specific button.

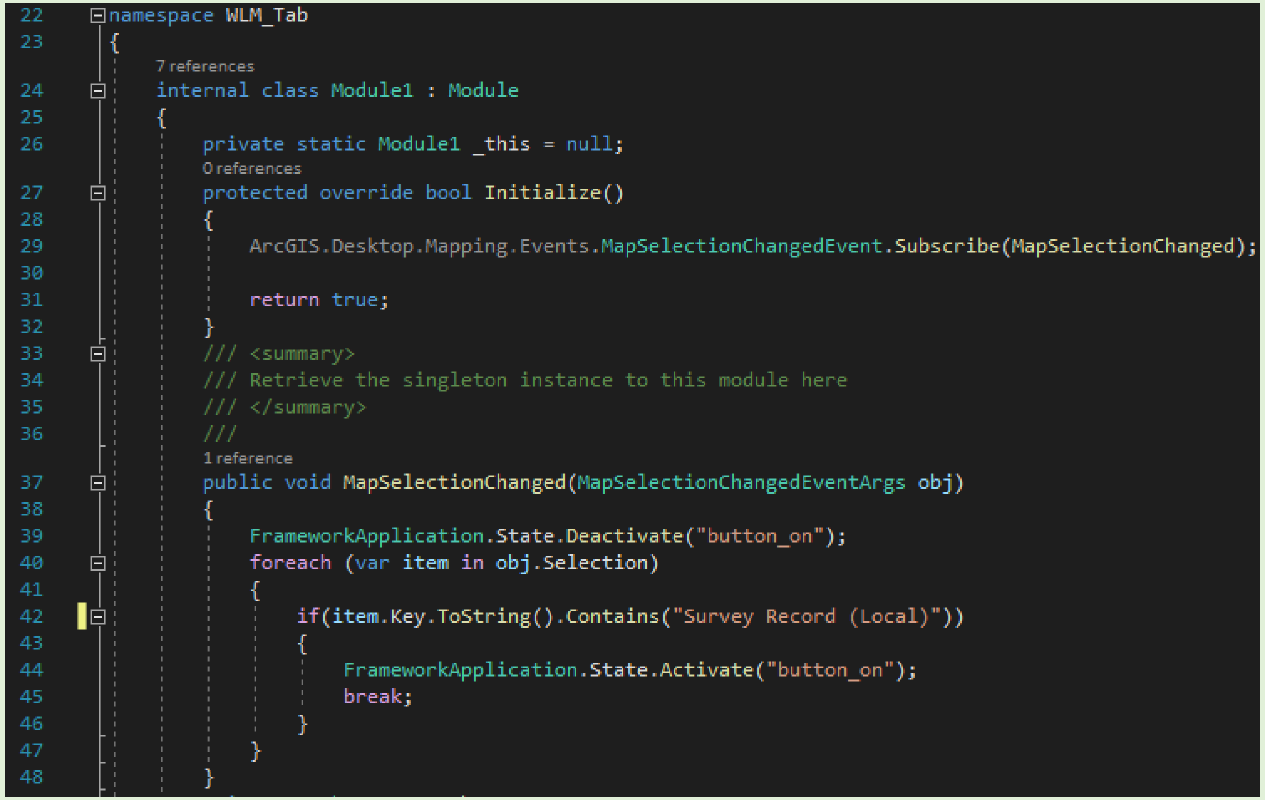
The code for the condition is inside the module1.cs so that it will work before being clicked.



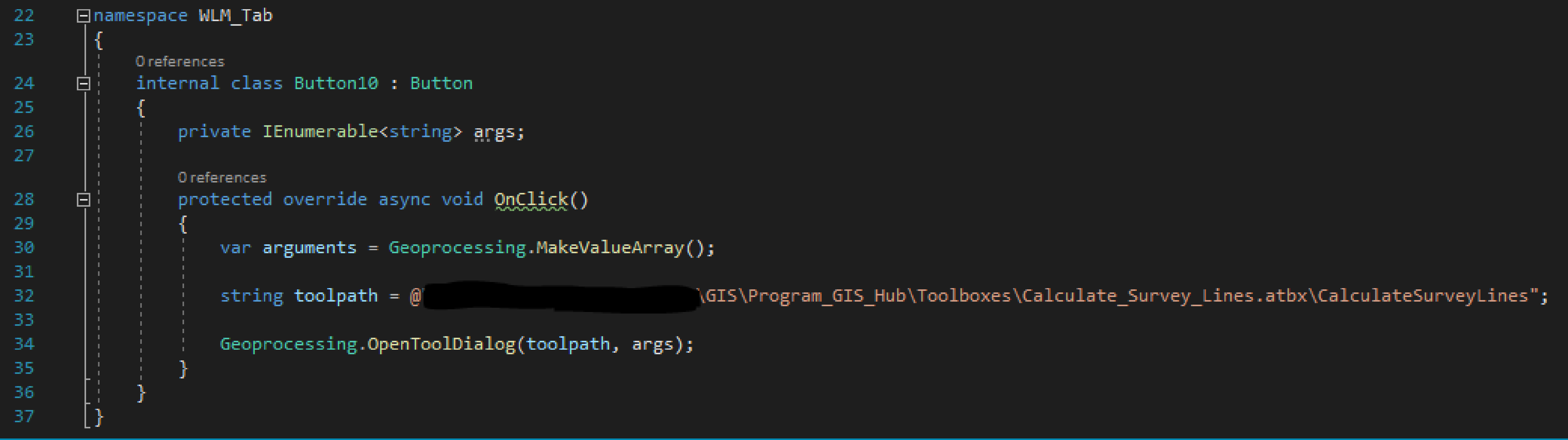
```
</AddInInfo>
  <conditions>
      <insertCondition id="button_on_selection" caption="button_on_selection">
          <state id="button_on"/>
      </insertCondition>
  </conditions>
<modules>
  <insertModule id="WLM_Tab_Module1" className="Module1" autoLoad="true" caption="Module1">
```



```
<button id="WLM_Tab_Button10" caption="Calculate Survey Lines" className="Button10" loadOnClick="true"
        smallImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/TableCalculateField16.png"
        largeImage="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/TableCalculateField32.png"
        condition="button_on_selection">
  <tooltip heading="Calculate Survey Lines Table">Calculate values to populate layout table for survey lines
  <disabledText /></tooltip>
</button>
```



```
namespace WLM_Tab
{
    7 references
    internal class Module1 : Module
    {
        private static Module1 _this = null;
        0 references
        protected override bool Initialize()
        {
            ArcGIS.Desktop.Mapping.Events.MapSelectionChangedEvent.Subscribe(MapSelectionChanged);

            return true;
        }
        /// <summary>
        /// Retrieve the singleton instance to this module here
        /// </summary>
        ///
        1 reference
        public void MapSelectionChanged(MapSelectionChangedEventArgs obj)
        {
            FrameworkApplication.State.Deactivate("button_on");
            foreach (var item in obj.Selection)
            {
                if(item.Key.ToString().Contains("Survey Record (Local)"))
                {
                    FrameworkApplication.State.Activate("button_on");
                    break;
                }
            }
        }
```



```
namespace WLM_Tab
{
    0 references
    internal class Button10 : Button
    {
        private IEnumerable<string> args;

        0 references
        protected override async void OnClick()
        {
            var arguments = Geoprocessing.MakeValueArray();

            string toolpath = @[redacted]\GIS\Program_GIS_Hub\Toolboxes\Calculate_Survey_Lines.atbx\CalculateSurveyLines";

            Geoprocessing.OpenToolDialog(toolpath, args);
        }
    }
}
```

Call to Script inside button

Most of the code here is generated by default from the SDK.

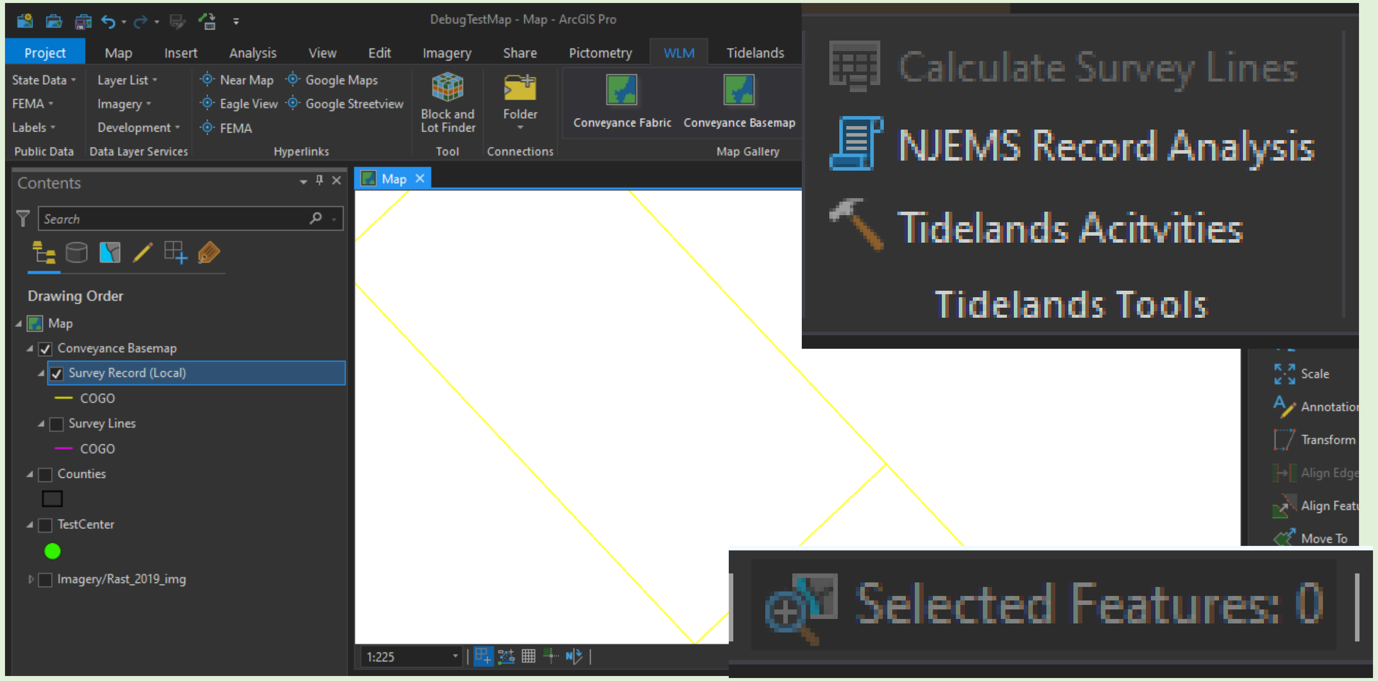
The button now gets connected to the Python script that was created earlier.

The Button that Doesn't Work

Disabled on startup, not after any sort of clicks.

Won't work unless a specific layer is inside the Table of Contents.

As useful as a screen door on a submarine.

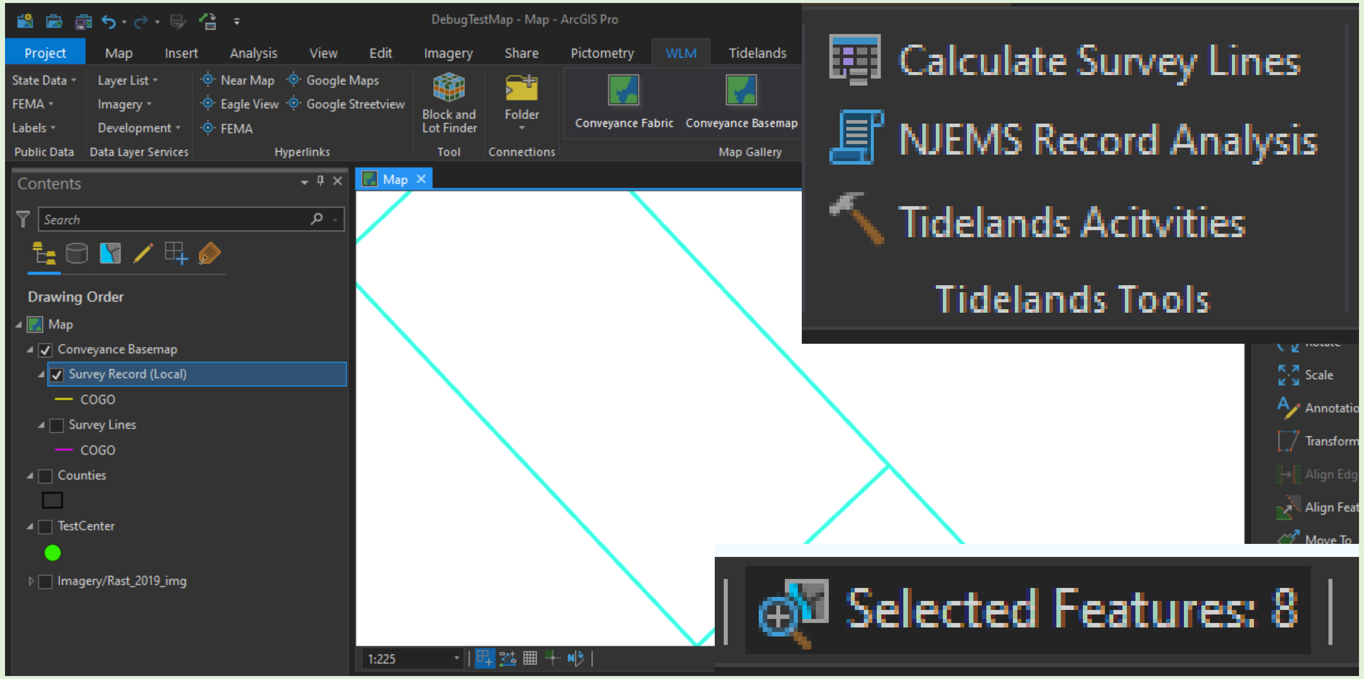


The Button that Does Work

Enabled after a selection is made.

Only enabled after a selection from a specific layer is made.

Only runs if all desired parameters are met.



Contact

Andrew C. Garcia

GIS Specialist II

NJDEP

Watershed & Land Management

- Andrew.Garcia@dep.nj.gov
- www.nj.gov/dep/wlm
- 609-940-5648

GitHub Project Link

- Like & follow us!

@newjerseydep

@nj.dep